AppleScript 帮助

概要

关于 AppleScript

在使用 Mac 时，您会发现您在重复地执行某些任务。使用 AppleScript，可以编写脚本和应用程序来自动执行重复性任务。

AppleScript 是一种功能强大的、通用的脚本语言，它内嵌在 OS X 中。可以使用 AppleScript 创建快捷、自动执行重复任务，甚至制作可以节省您很多时间的自定应用程序。AppleScript 可让您控制可脚本控制的应用程序。

AppleScript 脚本由类似英语的命令组成，这些命令在对象上执行操作。一些操作的示例有打开、关闭或获取名称。一些对象的示例有应用程序、窗口或偏好设置设置。

一个脚本能控制一个或多个应用程序。应用程序必须定义可以在脚本中使用的术语，才可以脚本控制。OS X 附带了几个可脚本控制的应用程序。

脚本也可以通过网络来控制应用程序，因此您可以在多台电脑上自动执行任务。

事实上，您已经在不知不觉中使用 AppleScript。该系统以及应用程序中的许多功能实际上就是 AppleScript 脚本，它们在应用程序中被调用。例如，在这个帮助系统中，每当您点按一个链接打开应用程序或文件时，那就是一个执行该操作的 AppleScript 脚本。

相关主题

AppleScript 如何工作
关于“AppleScript 编辑器”
关于 AppleScript 语言
互联网上的 AppleScript 资源
AppleScript Studio 开发环境

概要

关于“AppleScript 编辑器”

“AppleScript 编辑器”是一个简单易用的脚本编写应用程序，可以读取、编写、录制和存储 AppleScript 脚本。

“AppleScript 编辑器”的部分功能如下：

- 提供统一易用的界面
- 可以在脚本文本中查找和替换
- 自动绕排较长的脚本语句
- 可以执行多次撤销
- 提供快捷菜单脚本，用于快速编写脚本
- 支持应用程序服务
- 可让您使用“资源库调板”访问应用程序脚本

- 提供“脚本助理”来帮助您完成语句
- 提供结果历史记录和事件日志历史记录，以便您可以检查脚本的执行情况

相关主题

关于 AppleScript 语言
AppleScript Studio 开发环境

概要

创建 AppleScript 脚本

您无需知道任何关于编程的内容就可以利用 AppleScript。实际上，您不需要编写任何脚本。OS X 附带了成百上千个有用的 AppleScript 脚本，您可以使用它们，而且 Web 中也提供了成千上万个更多的 AppleScript 脚本。

如果您想更深入地使用它们，则可以获取现有的脚本，然后经过稍微的修改以满足您的需要。

当然，通过创建自己的脚本，也可以利用其大多数优点。学会如何编写脚本将使您能够创建有用的工具和节省时间的自动执行任务脚本，这将使您的工作更有效率。

您可以使用“AppleScript 编辑器”来修改脚本或创建您自己的脚本，它提供了一个简单的环境来供您编写脚本。“AppleScript 帮助”中的所有示例都是使用“AppleScript 编辑器”编写的。

- 打开“AppleScript 编辑器”。

  将以下文本键入到“AppleScript 编辑器”窗口。

  ```
  say "Welcome to AppleScript!"
  ```

  点按“运行”。

相关主题

关于“AppleScript 编辑器”
AppleScript Studio 开发环境

概要

AppleScript 如何工作

AppleScript 通过发送信息（称为“Apple 事件”）进行工作。运行脚本时，AppleScript 会将 Apple 事件发送到执行该命令的系统或应用程序。

在许多情况下，应用程序或系统将返回一个结果到 AppleScript。例如，如果脚本向应用程序发送请求以获得对象的名称，则应用程序会返回该名称。如果命令不能被执行，则该应用程序或 AppleScript 会返回错误信息。可以在“AppleScript 编辑器”中看到返回的结果。

通常，您无需知道“Apple 事件”的机制。AppleScript 在后台中执行所有这些操作。您仅需要编写脚本，然后运行它。

相关主题

关于 AppleScript
关于 AppleScript 语言

关于 Applet

存储为应用程序的脚本被称为 AppleScript 应用程序或 Applet。Applet 在打开时会自动运行，并将执行脚本中的命令。Applet 可以是仅包含几个命令的小函数，或者是包含成千上万行代码的复杂应用程序。

Applet 可以存储为可编辑或仅运行格式。可编辑 Applet 仍会包含脚本的文本，因此可以随时对它进行编辑和重新编译。若要编辑或更改 Applet，请将它拖到“AppleScript 编辑器”图标上。将脚本存储为仅运行格式时，可编辑文本部分将被移除，文件将不能再编辑或重新编译。

若要编辑存储为可编辑格式的 Applet，您可以在“AppleScript 编辑器”或其他 AppleScript 编辑应用程序中打开它。Applet 要求具有“on run”处理程序，当打开 Applet 时它将开始运行脚本。

- 本示例显示了“on run”处理程序。

```
on run
        --first command
        --second command
end run
```

相关主题

关于 Droplet
AppleScript 文件类型

关于已编译脚本

已编译的脚本是可编辑形式的 AppleScript 文件，您可以在“AppleScript 编辑器”或其他 AppleScript 编辑程序中读取和编辑它。

当打开已编译脚本时，它会出现在“AppleScript 编辑器”中，但不会自动执行任何命令。这将使您可以对脚本作任何修改，而不会实际运行它。然后您可以编译该脚本，这将读取文件中的所有文本并将它们转换为 AppleScript 代码。已编译的脚本可以运行。已编译脚本的文件扩展名为 .scpt。

将文件存储为已编译脚本时，文本仍与代码一起保留在该文件中。因此，您随时都可以再次编辑已编译脚本。

相关主题

关于 Applet
关于 Droplet
AppleScript 文件类型

关于 Droplet

Droplet 是当拖放文件或文件夹到其图标上时将会运行的一种 AppleScript 脚本。该脚本将对这些文件或文件夹执行操作。

与 Applet 一样，Droplet 也可以存储为可编辑或仅运行格式。可编辑 Droplet 仍会包含脚本的文本，因此可以随时对它进行

编辑和重新编译。若要编辑或更改 Applet，请将它拖到“脚本编辑程序”图标上。将脚本存储为仅运行格式时，可编辑文本部分将被移除，文件将不能再编辑或重新编译。

Droplet 要求具有“on open”处理程序，当打开 Droplet 时它将开始运行脚本。“on open”处理程序包含一个拖到 Droplet 中的项目列表。Droplet 脚本的一部分检查列表，还有部分对项目进行操作。除非 Droplet 仅设计用于对一个项目执行操作，否则它应该包含一个重复循环，该循环迭代列表中的所有项目，轮流对项目执行操作。

- 以下是带有“on open”和参数列表的 Droplet 处理程序的一个示例。

```
on open (theItemsDropped)
        --first command
        --second command
end open
```

相关主题

关于 Applet
关于已编译脚本
AppleScript 文件类型

概要

使用 AppleScript 字典

若要查明您可以使用哪些命令和对象来控制应用程序，您可以查看该应用程序的字典。字典包含了应用程序理解的一套术语。

可以在“AppleScript 编辑器”中查看应用程序的字典，方法是在“资源库”窗口中选择应用程序，或者将应用程序的图标拖到“AppleScript 编辑器”图标。

字典中的术语按照相关术语的分组（称为“术语组”）进行整理。命令和对象为粗体文本。

1. 在“AppleScript 编辑器”中，选取“窗口”>“资源库”。
2. 在“资源库”窗口中连按应用程序的名称。
3. 选择一个术语组。

   术语组中的术语及其说明会显示出来。

AppleScript 语言被整理成不同类别：

- 术语组：相关术语的分组。在“AppleScript 编辑器”字典中，术语组在第一栏内显示，用橙色字母“S”表示。
- 命令：请求操作或结果。命令也称为“事件”，显示在第二栏内，用蓝色字母“C”表示（也称为事件）。
- 类：可用脚本控制的对象。类的说明列出了类中存在的属性和元素。类显示在第二栏内，用淡紫色字母“C”表示。
- 属性：包含值的对象。属性显示在第三栏内，用淡蓝色字母“P”表示。
- 元素：包含于其他对象中的对象。例如，一个段落可能包含许多词。元素可能显示在第三栏内，用橙色字母“E”表示。

相关主题

类定义
命令定义
使用 AppleScript 来控制应用程序

打开

AppleScript 语言指南：命令参考
AppleScript 语言指南：类参考

概要

解决问题

如果您在使用 AppleScript 时遇到问题，请浏览以下主题，找到可帮助您解决问题的信息。

如果用户在 Choose From List 对话框中点按“取消”时出错

如果意外触发了文件夹操作

如果脚本不能识别磁盘

基本命令

类定义

类定义描述了可脚本控制对象的属性和元素。类可比作句子中的名词，类定义就像是名词的字典定义。

通常情况下，在 AppleScript 字典中类定义的开头为粗体形式的术语，后跟术语的简短描述（在同一行上）。

其后跟随一个任意元素或对象的列表，这些元素或对象属于或组成可脚本控制的对象。因为有些可脚本控制的对象包含元素列表，而有些则不包含，所以一个元素列表并不总是附带在类定义中。

类定义结尾带有可脚本控制对象的属性或质量的列表及它们的值。

属性条目的开头为该属性的名称，后跟：

- 属性的值储存时的数据格式的描述，例如整数（数字）、小数（两个数字的列表）、字符串（文本性数据）和布尔值（true 或 false）
- 可选类型指示符 [r/o]，表示属性的值为“只读”（只能读取不能更改）
- 属性的简短描述

如果对象有复数形式，则可以引用多个对象。

如果对象有元素列表，则它包含其他对象。可以在脚本中使用的元素名称以粗体显示。纯文本描述如何引用元素。

对象可能有属性。属性名称以粗体显示。纯文本会描述属性包含的值的类型，例如字符串。属性可以是一个常量值或一个常量列表。脚本可以获得并设定属性的值，除非属性在语法后带有“[r/o]”（只读）。

相关主题

命令定义
对象定义
AppleScript 命令
使用 AppleScript 字典

打开
AppleScript 语言指南：类参考

基本命令

命令定义

命令是 AppleScript 中执行操作的术语。AppleScript 字典中的命令显示在中间栏，用蓝色字母“C”表示。命令的定义包括命令用途及其语法的简短描述。语法是在脚本中编写命令的方法。语法可能包括几个部分。

粗体词是在语句中使用的术语。您必须按照字典中出现的术语准确地书写它们。

蓝色文本的词表示您提供的信息。例如，您可能要识别一个已定义的对象或变量。

命令有时会有可选部分，它们出现在方括号中。在语句中您不一定必须使用可选部分。

相关主题
类定义
使用 AppleScript 字典
事件定义
对象定义

打开
AppleScript 语言指南：命令参考

基本命令

事件定义

事件定义描述一个用来将操作指向可脚本控制对象的动词或命令。事件是对象可以完成或参与的操作或动词。

通常情况下，事件定义开头为术语，后跟术语的简短描述（在同一行上）。

事件定义后跟一列参数，命令执行时要用到这些参数。可选参数使用括号括起；而必需的参数则不用括号括起。

每个参数条目开头为参数的名称，后跟与参数一起使用的数据格式的描述，并以参数的一段简短描述结束。

命令的定义包括了命令做什么和它的语法的简短描述（带下划线的文字）。语法是在脚本中编写命令的方法。语法可能包括几个部分。

粗体词是在语句中使用的术语。您必须按照字典中出现的术语准确地书写它们。

纯文本的词表示您提供的信息。例如，您可能要识别一个已定义的对象或变量。

命令有时会有可选部分，它们出现在方括号中。在语句中您不一定必须使用可选部分。

命令作用于对象。对象的定义显示用来识别对象的语法，包括它的名称和简短描述（带下划线的文字）。对象的类决定了它的某些对象特征，例如可以配合该对象使用的命令。

相关主题
数据库事件
类定义
命令定义

对象定义
AppleScript 如何工作

基本命令

对象定义

命令作用于对象。在 AppleScript 字典中，对象的定义显示用来识别对象的语法，包括它的名称及简短描述（带下划线的文字）。对象的类决定了它的某些对象特征，例如可以配合该对象使用的命令。

如果对象有复数形式，则可以引用多个对象。

如果对象有元素列表，则它包含其他对象。可以在脚本中使用的元素名称以粗体显示。纯文本描述如何引用元素。

对象可能有属性。属性名以蓝色粗体显示。蓝色纯文本描述属性包含的值的类型，例如字符串。属性可以是一个常量值或一个常量列表。脚本可以获得属性的值和设定属性的值，除非属性在语法后带有“[r/o]”（只读）。

相关主题

使用 AppleScript 来控制应用程序
命令定义
类定义
告知对象执行什么操作
使用 AppleScript 字典

打开

AppleScript 语言指南：类参考
AppleScript 语言指南：属性

基本命令

If 语句

If 语句是仅在满足某些条件时才会执行的命令。每个 if 语句都包含一个计算结果为 true 或 false 的布尔表达式。if 语句中包含的命令仅在布尔表达式的值为 true 时才会执行。if 语句以 end if 结束。例如：

```
if amountReceived > 100 then
	set memberStatus to "special"
end if
```

在上述语句中，如果布尔表达式为 true（接收的值大于 100），则执行 if 语句中的命令。如果布尔表达式为 false，则不执行命令。

还可以构造带有 else 语句的 if 语句。这种组合语句称为 if-then-else 语句。例如：

```
if amountReceived > 100 then
	set memberStatus to "special"
else
	set memberStatus to "general"
end if
```

在上述语句中，如果第一个布尔表达式为 true（接收的值大于 100），则执行语句第一部分中的命令。如果第一个布尔表达

式为 false，则执行 else 下面语句第二部分中的命令。

可以将 if 语句互相嵌套，以便处理多个条件。

若要了解 AppleScript 中 if 语句的更多信息，请参阅“AppleScript 语言指南”。

相关主题

Repeat 语句
命令定义
在 AppleScript 中设定变量
常量值

打开

AppleScript 语言指南 - If

Repeat 语句

可以使用 repeat 语句根据条件多次执行任务。repeat 语句在一个循环中连续运行一组语句，直到符合某个条件并退出循环。可以将 repeat 语句相互嵌套，以便响应不同条件。

AppleScript 具有一系列 repeat 语句，可以响应不同条件。

---

repeat (forever)

在一个循环中不断重复。使用 exit 语句来退出循环。

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

---

repeat (number) times

重复指定的次数。

```
repeat 5 times
        -- statement
        -- statement
end repeat
```

---

repeat while

当指定的条件为 true 时重复。

```
set x to 1
repeat while (x < 5)
        -- statement
        -- statement
```

```
        set x to (x + 1)
end repeat
```

repeat until

不断重复，直到指定的条件为 true。

```
set x to 1
repeat until (x = 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

repeat with (variable) from (startValue) to (stopValue)

首先将变量设定为起始值，然后每循环一次就递增变量的值。当变量达到结束值时，重复将结束。

```
repeat with x from (5) to (25)
        -- statement
        -- statement
end repeat
```

repeat with (variable) in (list)

将变量设定为列表中的首个项目。循环每执行一次，变量每次都被设定为下一个项目。当使用了列表中的所有项后，重复将结束。

```
set myList to {234, 342, 681, 874}
repeat with x in myList
        -- statement
        -- statement
end repeat
```

若要了解 AppleScript 中 repeat 语句的更多信息，请参阅“AppleScript 语言指南”。

相关主题

If 语句
在 AppleScript 中设定变量
常量值

打开

AppleScript 语言指南 - repeat

基本命令

常量值

常量是 AppleScript 中包含预定值的保留字。AppleScript 包含许多常量，您可用来获取关于对象的信息，执行比较，或进行

算术运算。应用程序和其他对象可以定义各自的常量。这些常量在 value 类定义中已被定义。

若要查看常量的值，请将以下示例粘贴到“AppleScript 编辑器”，然后点按“运行”。点按“结果”查看常量的值。

结果也是一个常量。它是一个脚本常量，即命令返回的值。

```
get the current date
```

以下是常量的其他一些示例：

- 算术常量：pi
- 布尔常量：true、false
- 日期和时间常量：minutes、hours、days、weeks
- 字符串常量：return、space、tab
- 脚本常量：anything、current application、it、me、missing value、my、result

若要了解 AppleScript 中常量的更多信息，请参阅“AppleScript 语言指南”。

相关主题

在 AppleScript 中设定变量
对象定义
类定义
事件定义

打开

AppleScript 语言指南 - 常量

基本命令

在 AppleScript 中设定变量

变量是脚本中创建的用于储存值的一个容器。在创建变量时，可以为变量指定一个名称，又称为标识符。在运行脚本时，可以为变量赋值，访问值或修改值。变量仅在脚本运行时存在。当脚本关闭时，变量中的值会丢失。

- 这些示例显示如何指定唯一标识符并为它赋值。

```
set myVar to "153"

set x to "Quarterly Reports"

set x to false
```

若要查看值是如何储存在变量中的，请将这些示例粘贴到“AppleScript 编辑器”，然后点按“运行”。点按“结果”查看变量中储存的值。

若要了解 AppleScript 中变量的更多信息，请参阅“AppleScript 语言指南”。

相关主题

常量值

If 语句
Repeat 语句

打开

AppleScript 语言指南 - 变量

基本命令

指定应用程序的位置

OS X 中的应用程序被打包为“软件包”。软件包就是包含构成应用程序的那些文件夹和文件的特殊目录。因此，“path to”和“info for”命令应用于应用程序时会返回不同的结果。

如果您为应用程序使用“path to”命令，路径通常以“.app”结尾。在 Finder 中文件扩展名可能不会显示为文件名称的一部分，但您仍能使用扩展名来在脚本中识别文件。

如果您使用“info for”命令来获得关于应用程序的信息，则结果的“文件夹”属性通常是“true”。可以使用结果的其他属性来获得关于文件及其名称的更多信息，例如隐藏的扩展名、名称扩展名或显示名称。

相关主题

path to
info for
引用文件夹、文件和磁盘
choose application
Finder 脚本编写命令

Finder 命令

引用文件夹、文件和磁盘

AppleScript 可让您按名称（不包括路径）引用某些文件夹、文件和磁盘。AppleScript 可识别这些项目的位置，例如“文稿”文件夹、桌面、个人文件夹和启动磁盘。

- 本示例显示如何通过在脚本中使用其名称来引用特殊文件夹。

```
set x to the path to the desktop folder
```

相关主题

在脚本中引用 Finder
path to
info for
Finder 脚本编写命令

Finder 命令

在脚本中引用 Finder

若要使用 Finder 命令，请在 tell 语句中将命令指向 Finder，就像您将命令指向应用程序一样。

- 以下是在脚本中使用“tell”语句的一个示例。该 tell 语句指向“Finder.app”的窗口数函数的编号。

```
tell application "Finder.app"
        return the number of windows
end tell
```

相关主题

Finder
告知对象执行什么操作
引用文件夹、文件和磁盘
对象定义
Finder 脚本编写命令

打开

AppleScript 语言指南 - 对象指定符

Finder 脚本编写命令

以下是您可以配合 Finder 使用的一些命令：

activate

alias file

Alias list

application file

clean up

close

column

Computer-object

container

copy

count

data size

delete

desktop window

desktop

disk

document file

duplicate

eject

empty trash

exists

file

file information

find

Finder

Finder window

folder

get

icon view options

information window

Internet location file

item

list view options

make

move

moving folder window for

open

package

path to

preferences

preferences window

print

quit

引用文件夹、文件和磁盘

在脚本中引用 Finder

reveal

select

set

sort

trash

update

window

标准脚本编写附加命令

脚本编写附加添加了许多有用的命令，可供您在脚本中使用。

adding folder items to

beep

choose application

choose color

choose file

choose file name

choose folder

choose from list

choose URL

clipboard info

close access

closing folder window for

current date

delay

dialog reply

display

display dialog

do shell script

download

FTP item

get eof

info for

list disks

list folder

load script

mount volume

offset

open for access

open location

opening folder

path to

POSIX path

random number

read

round

run script

say

scripting components

set eof

set the clipboard to

set volume

store script

summarize

system attribute

time to GMT

upload

URL

write